

PLURAL ROBOT CONTROL SYSTEM

Publication number: JP5204428

Publication date: 1993-08-13

Inventor: TERADA TOMOYUKI; WATANABE ATSUSHI

Applicant: FANUC LTD

Classification:

- international: **B25J13/00; B25J9/16; B25J9/18; B25J19/06; G05B19/18; G05B19/19; G05B19/4155; B25J13/00; B25J9/16; B25J9/18; B25J19/06; G05B19/18; G05B19/19; G05B19/4155; (IPC1-7): B25J13/00; B25J19/06; G05B19/18; G05B19/19; G05B19/403**

- european: B25J9/16P4

Application number: JP19920035816 19920128

Priority number(s): JP19920035816 19920128

Also published as:



EP0582715 (A1)

WO9314910 (A)

US5561742 (A1)

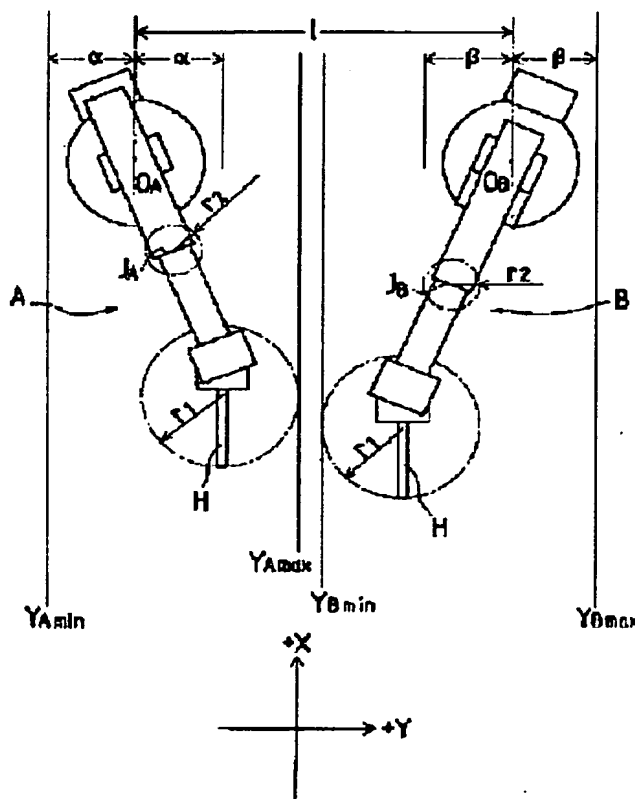
EP0582715 (A4)

EP0582715 (B1)

Report a data error he

Abstract of JP5204428

PURPOSE: To provide the control system of plural robots in which an operational program can be freely taught to each robot without predicting an interruption area, and the interruption can not be generated. **CONSTITUTION:** An operational occupancy area necessary for the execution of a command is searched for each command (Y_{Amax} , Y_{Amin}). When the interference with an occupancy area (Y_{Bmax} , Y_{Bmin}) stored by the other robot B at that time is not generated, the operation of the operational command is executed. When the pertinent interference is generated, a waiting is operated until the occupancy area of the other robot B is changed, and the interference is not generated. At the time of the completion of the execution, the occupancy area is stored according to the position of the robot at that time. The occupancy area is decided by the surface of a sphere including the wrist and hand H of the robot. Each robot executes the processing operation. It is not necessary to predict the occupancy area, and it is possible to freely teach the operational program to each robot.



Data supplied from the esp@cenet database - Worldwide

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-204428

(43)公開日 平成5年(1993)8月13日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 5 B 19/19		M 9064-3H		
B 2 5 J 13/00		Z 7331-3F		
	19/06	7331-3F		
G 0 5 B 19/18		C 9064-3H		
19/403		T 9064-3H		

審査請求 未請求 請求項の数8(全 8 頁)

(21)出願番号 特願平4-35816

(22)出願日 平成4年(1992)1月28日

(71)出願人 390008235

ファナック株式会社

山梨県南都留郡忍野村忍草字古馬場3580番地

(72)発明者 寺田 知之

山梨県南都留郡忍野村忍草字古馬場3580番地 ファナック株式会社内

(72)発明者 渡辺 淳

山梨県南都留郡忍野村忍草字古馬場3580番地 ファナック株式会社内

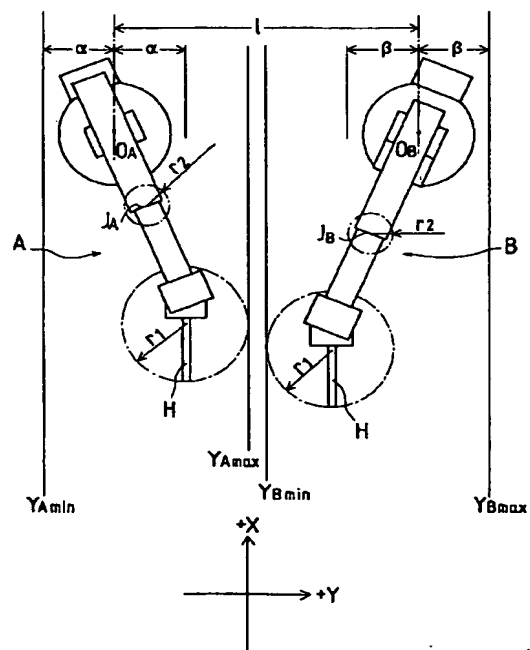
(74)代理人 弁理士 竹本 松司 (外2名)

(54)【発明の名称】 複数ロボット制御方式

(57)【要約】

【目的】 干渉する領域を予測することなく各ロボットに自由に動作プログラムを教示でき、かつ干渉が生じない複数ロボットの制御方式。

【構成】 動作指令毎、該指令を実行するに必要な動作占有領域を求める (Y_{Amax} , Y_{Amin})。他方のロボットBがその時点で記憶している占有領域 (Y_{Bmax} , Y_{Bmin}) と干渉しなければ、動作指令の動作を実行する。干渉するようであれば、他方のロボットBの占有領域が変わり干渉しなくなるまで待つ。実行が完了すると、ロボットのその時の位置によって占有領域を記憶する。占有領域は、ロボットの手首とハンドHを包含する球の表面によって決める。各ロボットは共にこの処理動作を実行する。予め干渉領域を予測する必要がなく自由に各ロボットに動作プログラムを教示できる。



1

【特許請求の範囲】

【請求項1】 複数のロボットを同時に動作させる作業におけるロボット制御方式において、各ロボット毎、各動作指令若しくは補間点がある毎にその指令によるロボット移動範囲によって決まる動作の占有領域を求め、他のロボットに対して記憶した占有領域に干渉しないときのみ上記求めた占有領域を当該ロボットの占有領域として記憶し、当該ロボットの当該動作を実行させ、実行後はロボットの位置に応じて占有領域を求め記憶する複数ロボット制御方式。

【請求項2】 ロボットのハンドと手首を覆うような球を及び肘関節を覆うような球をロボット形状の一部とし、該ロボット形状に基づいて上記占有領域を決める請求項1記載の複数ロボット制御方式。

【請求項3】 上記ロボットは垂直多関節ロボットで、ロボットのハンドと手首を覆うような球をロボット形状の一部とし、該ロボット形状に基づいて上記占有領域を決める請求項1記載の複数ロボット制御方式。

【請求項4】 各ロボットの据付平面上の設定された直線に各ロボットの形状を投影し、該直線上に占める各ロボット形状の区間によって上記占有領域を決める請求項1、請求項2若しくは請求項3記載の複数ロボット制御方式。

【請求項5】 複数のロボットを同時に動作させる作業におけるロボット制御方式において、隣り合う2つのロボットに対して夫々動作指令若しくは補間点がある毎にその指令によるロボット移動範囲によって決まる設定された直線上の他のロボット側における最大動作占有位置を求め、他のロボットに対して記憶した最大動作占有位置に干渉しないときのみ上記最大動作占有位置を記憶し、当該ロボットの当該動作を実行させ、実行後はロボットの位置に応じて上記最大動作占有位置を求め記憶する複数ロボット制御方式。

【請求項6】 上記設定された直線は各ロボット座標系における据付平面上の1座標軸である請求項4若しくは請求項5記載の複数ロボット制御方式。

【請求項7】 ロボットのハンドと手首を覆うような球を及び肘関節を覆うような球をロボット形状の一部とし、該ロボット形状に基づいて上記最大動作占有位置を決める請求項5記載の複数ロボット制御方式。

【請求項8】 上記ロボットは垂直多関節ロボットで、ロボットのハンドと手首を覆うような球をロボット形状の一部とし、該ロボット形状に基づいて上記最大動作占有位置を決める請求項5記載の複数ロボット制御方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、複数のロボットを同時に動作させ、その作業領域が一部重なるような作業を行わせるときのロボット制御方式に関する。

【0002】

2

【従来の技術】複数のロボットで共動作業を行うとき等、各ロボットの動作領域が重なり合うような場合、ロボット間の干渉を発生させないようにしなければならない。このような場合、従来行われている方法は、ロボット間で干渉が生じる可能性がある領域を予測し、この領域には同時に2台のロボットが作動しないように、ロボット間相互の通信を使用して、一方のロボットが該領域に侵入し作業中には、他方のロボットはこの領域に侵入しないように待機させるようにした制御が行われている。

【0003】

【発明が解決しようとする課題】上述した従来の方法では、ロボット相互の干渉する領域を予測しなければならず、この領域の予測が難しい。ロボット相互が干渉しないように安全を考え、この領域を大きく設定すれば、一方のロボットが該領域にある期間が長くなり、その結果、他方のロボットの待機時間も長くなることから作業効率は低下する。逆に、作業効率を上ようとして、干渉しない最小限の領域を予測し設定することは非常に難しくなる。また、ロボット相互の通信のタイミングを事前に設定する事が難しいため、多くの工数を必要とするという欠点がある。

【0004】そこで本発明の目的は、複数のロボットの相互に干渉する領域を予め予測する必要がなく、各ロボットに自由に動作プログラムを教示でき、かつ干渉が生じない複数ロボットの制御方式を提供することにある。

【0005】

【問題を解決するための手段】本発明は、複数のロボットに対して、夫々動作指令若しくは補間点がある毎にその指令によるロボット移動範囲によって決まる動作の占有領域を求め、他のロボットに対して記憶した占有領域に干渉しないときのみ上記求めた占有領域を当該ロボットの占有領域として記憶し、当該ロボットの当該動作を実行させ、実行後はロボットの位置に応じて占有領域を求め記憶するようにする。もしくは、占有領域を求める代わりに、隣り合うロボットにおいて設定された直線上における他のロボット側の最大動作占有位置を求め、該最大動作占有位置を上記占有領域にしてもよい。上記占有領域もしくは最大動作占有位置はロボットのハンドと手首を覆う球、さらには肘関節を覆う球をロボット形状として考え、該球が占める領域に基づいて決める。

【0006】

【作用】各ロボットに対して動作指令がある毎に、この動作指令により、ロボットが動作するために必要な占有領域もしくは上記最大動作占有位置が決まる。そのため、動作するときには、他のロボットが同様に占有領域もしくは最大動作占有位置として記憶している領域に干渉しないときのみ、その指令の動作を開始させる。また、1つの指令による動作を終了したときには、その時のロボットの位置（姿勢）によってロボットの占有領域

若しくは最大動作占有位置が決まるため、ロボットの位置に応じて占有領域若しくは最大動作占有位置を記憶しておく。これにより、複数のロボットが他のロボットの領域に干渉しないように動作を実行することができる。

【0007】ロボットの取り付け平面における動作領域は、ロボットハンド先端位置（もしくは手首位置）、肘関節位置及びロボットベースの大きさによって決まる。すなわち、ロボットをX-Y平面に設置した場合、ロボットアームが屈曲する肘関節のX、Y軸の位置、ロボット動作におけるロボットハンド若しくは手首のX、Y軸方向の位置とロボットベースの大きさによってロボットの動作領域若しくは上記最大動作占有位置が決まる。そのため、ロボットのハンドと手首を覆う球、肘関節を覆う球をロボット形状と想定し、このロボット形状が占める領域に基づいて占有領域若しくは最大動作占有位置を決めればよい。

【0008】

【実施例】図1は本発明の一実施例の説明図で、2台のロボットA、Bを据え付けた平面の垂直上方から見た図である。各ロボットのベース中心OA、OBを各ロボット座標系の原点とし、図1に示すように、横方向をY軸、縦方向をX軸、据え付け面に対して垂直方向をZ軸とし、ロボットA、Bのベース（固有の部分でロボットの動作によってX、Y軸方向には変動しない部分）の半径を夫々 α 、 β とする。2つのロボットA、Bの各動作状態における位置（姿勢）においてロボットA、Bの外形をX-Y平面、X-Z平面及びY-Z平面に投影したとき、投影した各平面でのロボット投影形状が1つでも干渉していなければ、ロボットA、Bは干渉しないことになる。すなわち、図1においてはX-Y平面にロボットA、Bの形状が投影され、干渉が生じていないから、この2第のロボットA、Bは干渉していないことになる。

【0009】さらにこの点を検討すると、ロボット形状が占めるX、Y、Z軸の区間において、1軸でも重なりが生じていなければ、2第のロボットA、Bは干渉が生じないことになる。図1の例で検討すると、ロボットA、BはX-Y平面に据え付けられているものであるから、Z軸方向の各ロボットが占める区間は必ず重なることになる。また、図1ではロボットA、Bの座標系のY軸が一致する状態に配置しているから、ロボットA、BのX軸上に占める区間は必ず重なりが生じる。しかし、Y軸上においてはロボットA、Bの位置姿勢によって重なりが生じる場合と生じない場合がある。すなわち、図1のようにロボットA、Bを配置したときには、ロボットA、BのY軸上に占める区間が重ならなければ、ロボットA、Bが干渉しないことを意味する。また、ロボットA、Bの据え付け位置をX軸もしくはY軸が一致しないように配置した場合には、X軸上、Y軸上で各ロボットの投影部分が重なりあう場合と重なり合わない場合が

生じ、X軸もしくはY軸上のどちらか一方で重ならないことが判明すれば干渉がないことがわかる。この場合、座標系原点OA、OB間の距離が短い軸の方が、重なり合うことが多くなるから、距離が大きい方の軸で干渉をチェックした方がよい。さらに、干渉を精度よくチェックするには、2台のロボットのロボットベース中心（OA、OB）間を結ぶ線上にロボット形状を投影し、この線上で重なるか否かをチェックするようにすればよい。

【0010】本実施例では、図1に示すようにロボットを配設し、Y軸上において重なりが生じるか否かをチェックすることによってロボットA、Bの干渉をチェックする例を示す。この場合、ロボットベース部分はロボットの位置や姿勢によって変動せず、一定であるから、ロボットA、BがY軸上に投影されて変動する部分を決定する部分はロボットアーム先端に取り付けられたハンド先端位置が一般的にである。しかし、関節部分においては、ハンド先端よりも大きなY軸の値を取り、これによりY軸上に占めるロボットの投影区間が変動することがある。ロボットベース上で回転、旋回もしくはZ軸方向に伸縮する関節部分は、ロボットベースが占めるY軸上の値を越えることはないが、ロボットアームの途中に設けられている関節によっては、その関節が屈曲、旋回してハンド先端よりもこの関節の部分がY軸上大きな値を取り、Y軸上においてロボットが占有する区間を長くする場合がある。以下このような関節を肘関節という。

【0011】そこで、各ロボットA、Bの動作占有領域を決定するには、ロボットハンドHの位置とロボットベース半径 α 、 β 、及び肘関節の位置を考慮すればよい。そこで本実施例においては、ロボット手首及びハンドHは回転、旋回するものであるから、また肘関節は屈曲旋回するものであるから、手首及びハンドHを包含するような球及び肘関節の端部をすべて包含する球を求め、このこれらの球の表面がY軸上で占める値の大小によって、ロボットの動作占有領域の境界線を決める。この球の表面位置はこの球の中心位置と球の半径が分かればよく、ハンド部の球の中心位置（以下この中心位置をハンド代表点位置という）と半径はハンドHの種類とロボット位置（姿勢）によって求めることができる。また、肘関節における球の中心位置は、ロボットの位置（手首中心位置）を求める際に関節の位置として求められ、ロボットアームの外形よりこの球の半径は決まる。

【0012】図1の例では、アームの途中にロボットの占有領域を決めるに当たり影響のある肘関節が1つある例を示しており、該肘関節JA、JB部の球の半径を r_2 、ハンド部の球の半径を r_1 としている。そして、図1の状態では、ロボットAの占有領域はY軸方向のベース半径 α によって最小値 Y_{Amin} が決まり、最大値は肘関節JA部の球の表面のY軸値よりハンド部の球の表面位置のY軸値（ハンド代表点位置のY軸座標位置に球の半径 r_1 を加算した位置）が大きいので、この位置が占有

領域の最大値 Y_{Amax} として決まる。また、同様にロボットBの占有領域は、図1に示すように、最大値 Y_{Bmax} はベース半径 β によって決まり、最小値 Y_{Bmin} はハンド部の球の表面によって決まる。そして、この2つの占有領域が干渉しなければ、2台のロボットは同時に動作させることができる。そして、干渉するか否かは、2台のロボットの原点OA,OB間のY軸方向の距離を L とすると、次ぎの1式の関係式が成立すれば、干渉しないことになる。

$$[0013] \quad Y_{Amax} - Y_{Bmin} < L \quad \cdots (1)$$

(なお、 Y_{Bmin} は負の値である)

図2、図3は複数のロボットを同時に動作させて作業を行わせるときの制御システムの構成図で、従来から実施されている構成と同じものである。図2は1つのロボットコントローラC1によって複数のロボットを制御する場合の構成を示している。この図2では2台のロボットA、Bしか接続されていないが、さらに他のロボットも接続されるものである。また、図3に示す例は、ロボットコントローラC2、C3間を通信によって接続するタイプのものである。また、図3では各ロボットコントローラC2、C3に夫々1台のロボットA若しくはBが接続されているのみであるが、夫々他のロボットを接続し複数のロボットを制御できるものである。

[0014] また、図4はロボットコントローラの構成を説明するブロック図であり、このロボットコントローラの構成も従来と同様の構成を有するもので、周知のものである。プロセッサ(CPU)10には、システムプログラムを記憶するROM11、各ロボットに対する教示プログラムや各種設定値、パラメータ等を記憶するRAM12、各ロボットのサーボ機構(サーボモータ)を駆動するサーボモジュール13、各ロボットの各種センサやアクチュエータとの信号を入出力するI/Oモジュール14がバス15で接続されている。なお、具体的な各要素の作用は周知のものであるので説明を省略する。

[0015] 次に、図5に示すフローチャートと共に、本実施例の動作を説明する。なお、図5に示すフローチャートは図1に示すロボットAに対してすでに教示された動作プログラムに基づく動作処理のフローチャートである。まず、各ロボットに取り付けたハンドHの各種パラメータを設定するときに、ロボット手首及びハンドHを包含するような上記球の半径 r_1 、及び該球の中心位置のハンド代表点位置を規定するための手首中心位置(ロボットへの動作指令位置)に対するオフセット量を設定すると共に、肘関節JA、JB部の球の半径 r_2 及び各ロボットのベース半径(図1の場合は α 、 β)さらに干渉が生じるロボットの座標原点間の干渉チェックを行う軸方向の距離を設定する。図1に示す実施例の場合にはY軸方向の距離 L を設定する。

[0016] そして、ロボットAに対して動作指令を入力すると、ロボットコントローラのプロセッサ10は、

まず、ロボットAの現在位置(手首中心位置)を求めると共にその過程で肘関節JAの位置も求める。そして上記求められたロボットAの現在位置(手首中心位置)とハンド代表点位置に対する設定オフセット量より、ハンド代表点位置を求める。図1に示す実施例では、ロボット干渉のために必要なデータはY軸の位置のみであるから、このフローチャートでは肘関節位置、代表点位置としてY軸の値 Y_j 、 Y_h を求めている(ステップS1)。

[0017] 次に、ロボットAに対して設定されているベース半径 α と、ハンド代表点位置のY軸値 Y_h に設定された球の半径 r_1 を加算した値、及び肘関節JAのY軸値 Y_j に設定された肘関節部に対する半径 r_2 を加算した値の3つの値を比較し一番大きい値を動作領域の最大値 Y_{Amax} としてレジスタに記憶する。また、ハンド代表点位置のY軸値 Y_h から球の半径 r_1 を減じた値、肘関節JAのY軸値 Y_j に球の半径 r_2 を減じた値とベース半径を負にした値 $-\alpha$ と比較し1番小さい値を動作領域の最小値 Y_{Amin} としてレジスタに記憶する(ステップS2)。なお、図1に示すような実施例では、ロボットAのY軸負の方向で干渉が生じないからこの領域最小値 Y_{Amin} を求める必要はないが、図1において左側にさらにロボットが設置されて干渉が生じる場合にはこの領域最小値 Y_{Amin} が必要になってくる。なお、このレジスタに記憶する最大値、最小値の値は、図3のタイプのシステムで制御する場合には、他方のロボットコントローラに送信し記憶させる。

[0018] そして、教示された動作プログラムより1ブロック(プログラムの1行、または1動作指令)を読み(ステップS3)、該指令が動作指令か否か判断し(ステップS4)、動作指令でなければ終了指令があるか否か判断する(プログラム上に指令されているか、若しくはオペレータが終了指令を入力しているかを判断する)(ステップS12)。終了指令でなければ、指令された他の処理を実行し(ステップS13)、ステップS4に戻る。また、動作指令であると、この指令されたロボット位置より求められるハンド代表点位置のY軸値 Y_h' 、肘関節JAのY軸値 Y_j' をステップS1と同様に求め(ステップS5)、ステップS2と同様に、この動作指令を実行したときの占有領域の最大値 Y_{Amax}' 、最小値 Y_{Amin}' を求める(ステップS6)。そして、この最大値 Y_{Amax}' 、最小値 Y_{Amin}' で決まる領域と他のロボットの占有領域としてレジスタに記憶している領域が干渉するか否かを判断する。図1に示す本実施例の例では、ロボットAの領域最大値 Y_{Amax}' とロボットBのその時点におけるレジスタに記憶した占有領域の最小値 Y_{Bmin} 及び、設定原点間距離 L により次の2式が成立するか否かを判断し干渉チェックを行う(ステップS7、S8)。

$$Y_{Amax}' - Y_{Bmin} < L \quad \cdots (2)$$

上記2式が成立せず、干渉が生じる場合にはステップS7、S8の処理を繰り返し実行し、動作を一時中断し待機する。ロボットBが動作を行いレジスタに記憶するロボットBの占有領域が変更され、上記2式が成立するようになると、まず、この1動作の動作前と後の領域最大値 Y_{Amax} 、 Y_{Amax}' の大きいほうを占有領域の最大値 Y_{Amax} としてレジスタに記憶する。また1動作前と後の領域最小値 Y_{Amin} 、 Y_{Amin}' の小さいほうを占有領域の最小値 Y_{Amin} としてレジスタに格納し、動作占有領域を確保し(ステップS9)、この指令に対する動作を実行する(ステップS10)。すなわち、ある点から次の点に移動する場合には、動作占有領域が変動するが、この動作を実行するにはこの動作過程での最大の領域を確保しておけばよいからステップS9の処理を実行するものである。

【0019】動作の実行が終了すると、動作終了後のロボットの位置で占める占有領域である Y_{Amax}' 、 Y_{Amin}' をレジスタに記憶し(ステップS11)、ステップS3に戻り次のブロックの処理を行う。

【0020】上記図5に示すフローチャートでは、ロボットAに対する処理のみを記載したが、ロボットBに対しても同様の処理が行われ、ロボットAとBの相互の干渉が生じないように各ロボットは動作を実行することになる。

【0021】また、上述した動作はプログラムによる動作の実行についてのべたものであるが、ロボットに対して教示する場合にも同様な処理が実行される。それは、一方のロボットに対して教示しているときに他方のロボットに対しても教示が行われる場合もあり、また他方のロボットが動作中に教示が行われる場合もあるからである。この教示動作の場合は、図5のフローチャートでステップS3、S4の処理が、教示入力ありか否かの判断処理に代わる点と、ステップS12の処理が教示終了指令が入力されたか否かの判断処理に代わる点のみが相違し、他の処理は図5の処理と同一である。

【0022】さらに、上記実施例では、動作指令単位で干渉チェックを行うようにしたが、補間計算を行った後、各補間点毎に干渉チェックを行うようにしてもよい。すなわち、図5のステップS4で動作指令と判断されると、該指令に基づき補間計算を行い各補間点を求め、各補間点に対応するハンド代表点位置 Y_h' 、肘関節位置 Y_j' を求め、最初の補間点よりステップS6～S11の処理を順次繰り返し行い、動作指令の終点まで達すると、ステップS3に戻る処理を行うようにすればよい。

【0023】なお、上記実施例では図1に示すように2台のロボットの干渉を防止した制御について述べたが、さらに多くのロボットを併設したときも同様に干渉を防止する制御を行うものである。例えば、図1において、ロボットAの左側のY軸線上にロボットCを据え付けた

場合には、ロボットAと、Cの干渉チェックが同様に行われる。この場合、図1の実施例に置けるロボットAとロボットBの関係がロボットCとロボットAの関係に置き代わるだけである。また、ロボットDがロボットBの右側に設置され他場合にも、図1におけるロボットAとロボットBの関係が、ロボットBとロボットDの関係に置き代わるだけである。

【0024】上記実施例では、肘関節JA、JBの位置も占有領域決定の1パラメータとして取り扱ったが、垂直多関節ロボットの場合には、肘関節JA、JBはZ軸方向に屈曲し肘関節JA、JBがハンド位置よりY軸上において原点OA、OBより遠い位置になる場合は非常に少ない。そのため、ロボット教示プログラムに上述したような肘関節JA、JBがハンド位置よりY軸上において原点OA、OBより遠い位置になるような動作を教示しないような場合には、肘関節JA、JBの位置を占有領域の判断パラメータから除外してもよい。

【0025】また、上記実施例では、各ロボットの占有領域として座標系におけるY軸値を利用した。すなわち、各ロボットの座標系において、ロボット座標系原点間の距離が1番長い座標軸における各ロボットが占有する値によって各ロボットの占有領域を判断するようにした。しかし、必ずしも座標軸にロボット形状を投影してこの軸の値によって占有領域を決め干渉チェックを行うようにしなくてもよい。1番干渉を発見できるような任意の直線(据付面上における直線)に各ロボット形状を投影し占有領域を求めこの領域が干渉するか否かでロボットの干渉チェックを行うようにしてもよい。例えば、上記直線としてロボットA、Bの座標原点OA、OBを結ぶ直線とする。この場合には座標系の1軸を干渉判断の直線としたときよりも精度が高く干渉チェックができる。また、3台以上のロボットを同一直線上に配列した場合には、その直線を干渉チェックの直線とすればよく、同一直線上に配列しない場合には、各ロボットの座標原点を結ぶ折れ線を直線で近似して該直線を干渉チェックの直線とすればよい。

【0026】さらに、上記実施例では、ロボットベースの大きさによっても占有領域を規定するようにしたが、各ロボットの最大動作範囲内に他方のロボットのロボットベースが位置付けられないように配置した場合(例えば、図1においてロボットAのアームを最大に伸ばし上記球の表面がロボットBのベース領域(半径 β の範囲内)に入らない場合)には、占有領域を決める上でロボットベースを考慮する必要はない。

【0027】

【発明の効果】本発明では、ロボットの位置(姿勢)及びロボットの1動作において、ロボットが占有する領域を各ロボット毎求め、一方のロボットが1動作を実行しようとするときに他方のロボットが占有している領域に干渉するような場合には、干渉が生じなくなるまで待つ

て動作を開始するようにしたから、予め干渉する領域の予測を行う必要がなく、かつ、ロボット相互の通信のタイミングを事前に設定する必要がないから、各ロボットに対して自由に動作プログラムを教示実行させることができる。これにより、ロボットを動作させるための工数が少なくなる。

【図面の簡単な説明】

【図1】本発明の方式を実施する一実施例の説明図である。

【図2】1台のロボットコントローラで複数のロボット

を制御するシステム構成図である。

【図3】ロボットコントロール間を通信で結合し、複数のロボットを制御するシステム構成図である。

【図4】ロボットコントローラの要部ブロック図である。

*

*【図5】本発明の一実施例のロボット制御処理のフローチャートである。

【符号の説明】

C1～C3 ロボットコントローラ

A, B ロボット

H ハンド

α, β ロボットベースの半径

r 球の半径

L ロボットの座標系原点間の距離

Y_{Amax} ロボットAの占有領域の最大値

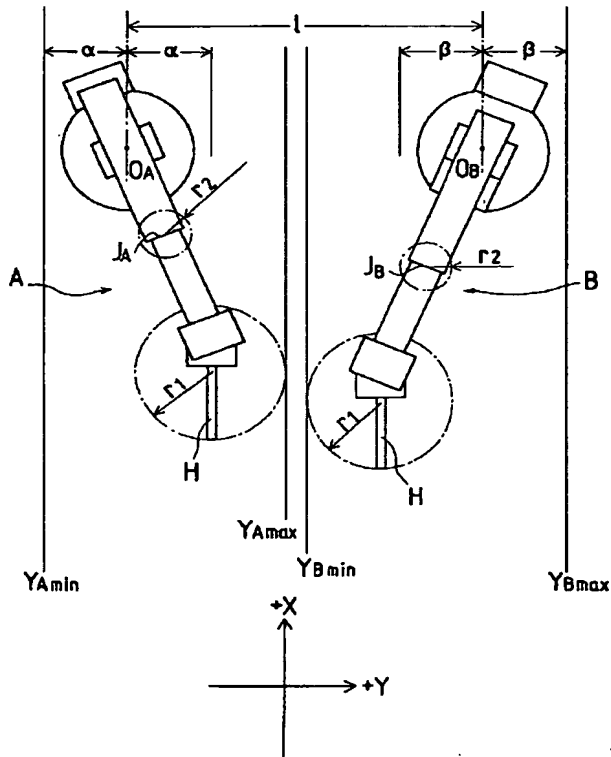
Y_{Amin} ロボットAの占有領域の最小値

Y_{Bmax} ロボットBの占有領域の最大値

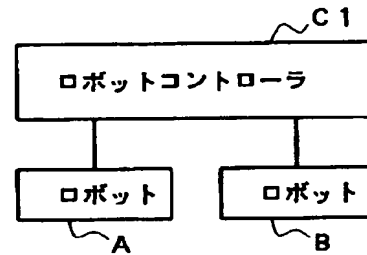
Y_{Bmin} ロボットBの占有領域の最小値

J_A, J_B 肘関節

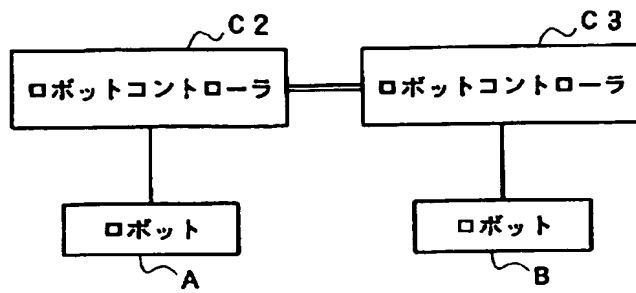
【図1】



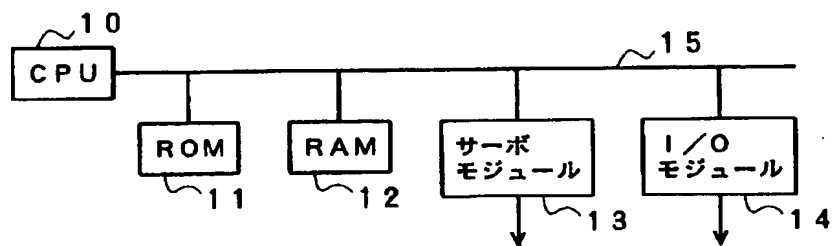
【図2】



【図3】



【図4】



【図5】

